

# JetScape Directory- structure

LongGang Pang

# Current status

- Move all your files to `src/framework`
  - Stop any developments that are outside the new root directory

Previous Project Root:[JETSCAPE-COMP/](#)

```
JETSCAPE-COMP $ls  
3rdparty/           INSTALL.txt      README.md        build/          interface/       test/  
CMakeLists.txt     LICENSE          RedmineTest.rtf  external packages/ src/      wrappers/
```



Current Project Root:[JETSCAPE-COMP/src/framework](#)

```
framework $ls
3rdparty/           Modules/          cmake_clean.sh    hydro/          reader/
CMakeLists.txt      README.md        graph_fancy.py  jet/           setup.csh   src/
JetScapeDoxy.conf  build/           gtl/             jetscape_init.xml  setup.sh   test/
```

# Future structure

- JETSCAPE-COMP/src/framework/\* will be move to JETSCAPE-COMP/
- Use conventions for each directory

```
JETSCAPE-COMP $ls
3rdparty/          Modules/
CMakeLists.txt    README.md
JetScapeDoxy.conf build/           cmake_clean.sh
                           graph_fancy.py
                           gtl/           hydro/
                           jet/           jetscape_init.xml
                           reader/
                           setup.csh
                           setup.sh           src/
                           test/
```

- 3rdparty/: store 3rdparty libraries that we use but do not touch
  - framework \$ls 3rdparty/  
googletest/
  - hydro\_from\_external\_file/
  - trento/
  - vhle/
- Modules/ -> cmake/: store \*.cmake file for looking for libraries

```
Modules $ls
FindCERNLIB.cmake   FindGSL.cmake      FindHEPMC.cmake     FindPythia6.cmake   FindPythia8.cmake
FindROOT.cmake      FindROOT.cmake.root
```

# Functions that have clean meaning

```
JETSCAPE-COMP $ls
3rdparty/          Modules/
CMakeLists.txt     README.md
JetScapeDoxy.conf  build/
                                         cmake_clean.sh    hydro/
                                         graph_fancy.py   jet/
                                         gtl/             jetscape_init.xml reader/
                                         setup.csh       setup.sh
                                         src/
                                         test/
```

- hydro/ jet/ reader/
- Maybe it is better to move files in these directories to src/

# Unit-tests and program examples

```
JETSCAPE-COMP $ls
3rdparty/          Modules/
CMakeLists.txt     README.md
JetScapeDoxy.conf  build/
                           cmake_clean.sh
                           graph_fancy.py
                           gtl/
                           hydro/
                           jet/
                           jetscape_init.xml
                           reader/
                           setup.csh
                           setup.sh
                           src/
                           test/
```

- We should separate unit-tests from program tests
- Unit-tests: test functionality of fine grind implementations like one function, one class or one module
- Program tests: like [main0.cc](#), [main1.cc](#), ... [main\\*.cc](#) in [pythia/examples/main\\_auau200.cc](#) or [pythia/examples/auau200/main.cc](#)
- Rename [test/](#) -> [examples/](#) like in [pythia](#)
- [MUSICTest.cc](#), [brickTest.cc](#), [testJetScape.cc](#) [test\\*.cc](#) in [framework/src/](#) should be moved to [framework/examples](#) or [framework/src/test/](#)

# Unit-tests in hydro/

```
hydro $ls  
CMakeLists.txt          fluid_dynamics.cc      fluid_dynamics.h      linear_interpolation.h  realtype.h  
test/
```

```
test $ls  
CMakeLists.txt          fluid_dynamics.cc      linear_interpolation.cc
```

```
// framework/hydro/test/linear_interpolation.cc  
  
#include "../linear_interpolation.h"  
#include "gtest/gtest.h"  
  
TEST(LinearInterpolationTest, TEST_TRUE){  
    EXPECT_EQ(0.5, linear_int(0.0, 1.0, 0.0, 1.0, 0.5));  
}  
  
// test code when the type of y is int  
TEST(LinearInterpolationTest, TEST_INT){  
    EXPECT_EQ(0, linear_int(0.0, 1.0, 0, 1, 0.5));  
}
```

# Unit-tests and CMakeLists.txt in src/

```
// framework/src/test/JetScapeInitial.cc

#include "../InitialCondition.h"
#include "gtest/gtest.h"

TEST(JetscapeInitialTest, TEST_SAMPLE){
    auto ini = JetScapeInitial("auau200", 0, 5, 10, 0.2);

    EXPECT_EQ(ini.entropy_density_distribution_.size(), 10000);
    double mul = ini.info_.total_entropy;
    ASSERT_TRUE(mul >= 100 && mul <= 150);
}

1 ##### // framework/src/test/CMakeLists.txt
2 # Unit Tests
3 #####
4 add_executable(test_jetscape_ini JetScapeInitial.cc)
5
6 # Standard linking to gtest stuff.
7 target_link_libraries(test_jetscape_ini gtest gtest_main)
8 # trento asks for Boost and HDF5 libraries
9 target_link_libraries(test_jetscape_ini JetScape)
10
11
12 # trento asks for Boost and HDF5 libraries
13 #target_link_libraries(test_jetscape_ini libtrento ${Boost_LIBRARIES} ${HDF5_LIBRARIES})
14 # This is so you can do 'make test' to see all your tests run, instead of
15 # manually running the executable runUnitTests to see those specific tests.
16 add_test(NAME test1 COMMAND test_jetscape_ini)
```

# Build directory

- Put all **executables** to build/ directory
- cd build/
- cmake ..
- make or make test

```
build $ls
3rdparty/
test_fluid_dynamics
CMakeCache.txt
test_jetScape_ini
CMakeFiles/
test_linear_interpolation1d
CTestTestfile.cmake
          Makefile
          brickTest
          cmake_install.cmake
          gtl/
          hydro/
          jet/
          lib/
          reader/
          readerTest
          src/
          test/
          testJetScape
          trento
```